

# Stegräknare

Programmering av inbyggda system  
Mjukvaruutvecklare inbyggda system 2016

EJ FÄRDIG RAPPORT

Dennis Bunne  
Simon Karlsson  
Johan Kämpe

# Sammanfattning

Skrivs när rapport är klar.

# Innehållsförteckning

<b>Sammanfattning</b>	<b>2</b>
<b>Innehållsförteckning</b>	<b>3</b>
<b>1 Inledning</b>	<b>4</b>
1.1 Syfte	4
1.2 Bakgrund	4
1.2.1 Stegräknare	4
1.2.2 I <sup>2</sup> C	4
1.2.3 X-Y-Z	4
<b>2 Genomförande och resultat</b>	<b>5</b>
2.1 Använd programvara	5
2.2 Komponentlista	6
2.3 Utveckling av stegräknaren	7
2.3.1 Prototyp 1	7
2.3.2 Prototyp 2	9
2.4 Stegräknarens funktion	12
2.5 Kod	13
2.5.1 Använda icke-standardbibliotek	13
2.5.2 Globala variabler	13
2.5.3 Definitioner / macron	14
2.5.4 Funktioner	14
<b>3 Referenser</b>	<b>15</b>
<b>4 Slutsats</b>	<b>15</b>

# 1 Inledning

## 1.1 Syfte

Syftet med projektet är att tillverka en bärbar, batteridrivnen stegräknare. Stegräknaren ska vara modulär, dvs det ska gå att ansluta och ta bort enheter, display, WiFi-modul m.m.

Initialt används en display för att visa antal steg. Stegen ska avläsas med hjälp av en accelerometer, eller alternativt med en vibrations sensor.

Stegräknaren kan senare utökas med hastighetsmätare, tidsräkning, ESP-modul eller bluetooth-modul för att skicka data m.m, om funktionen med stegräkning fungerar korrekt.

## 1.2 Bakgrund

### 1.2.1 Stegräknare

En stegräknare är en, vanligtvis portabel, enhet som mäter hur många steg en människa tar. Den kan användas vid exempelvis löpning eller promenad. Eller för att se hur många steg en person tar under en dag. Stegräknare kan också kallas pedometer.

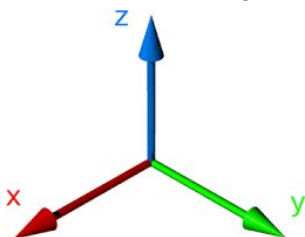
### 1.2.2 I<sup>2</sup>C

I<sup>2</sup>C, Inter-Integrated Circuit, är ett sätt att koppla enheter till moderkort, inbyggda system, mobiltelefoner eller andra enheter. Kommunikation sker via två ledare, SDA (Serial Data Line, datasignal) och SCL (Serial Clock, klocksignal). Flera I<sup>2</sup>C-enheter kan anslutas till samma SDA- och SCL-ledare.

### 1.2.3 X-Y-Z

X, Y och Z används inom projektet för att beskriva riktningar i en tredimensionell miljö. MPU-9250-enheten har sensorer för acceleration i dessa tre riktningar. När enheten förs åt ett visst håll mäts rörelsens acceleration i denna riktning.

X- Y- och Z-riktningar.



## 2 Genomförande och resultat

### 2.1 Använd programvara

Vid utveckling av stegräknaren har följande programvara använts

Program	Beskrivning
<b>Atmel Studio 7.0</b>	IDE för att skriva kod och kompilera till Atmel MCU.
<b>Git</b>	Versionshanteringsverktyg
<b>GitKraken</b>	Grafiskt gränssnitt för Git
<b>Fritzing</b>	Program för att skapa kretsscheman
<b>Microsoft Word</b>	Dokumentering
<b>AVR Dudess</b>	Program för att ladda upp koden till MCU i form av en HEX-fil.
<b>Autodesk Fusion 360</b>	CAD-program för att skapa ett hölje för produkten

## 2.2 Komponentlista

För att bygga stegräknaren har följande hårdvara använts:

Artikel	Beskrivning
<b>Atmel ATmega328-PU DIP-28N 8-bit MCU</b>	Microprocessor.
<b>SparkFun MPU-9250</b>	Enhet som mäter acceleration i X, Y och Z-riktning. Samt gyroskop, temperatur och kompass-funktionalitet.
<b>Micro OLED Display SSD1306 128x64</b>	Display med upplösningen 128x64.
<b>AVR pocket programmer</b>	Enhet för att programmera Atmega-MCU
<b>DIL-hållare 28-pin 0.3"</b>	Sockel för Atmega-MCU
<b>Resistor 4,7k<math>\Omega</math>, 2 st</b>	Pull up-resistorer för I <sup>2</sup> C-kanaler.
<b>Kristalloscillator 16 MHz</b>	Kopplas till MCU för att öka klockfrekvensen från 1 MHz till 16 MHz
<b>Kondensator 22pF, 2 st</b>	Ger jämn ström till Oscillatorn
<b>Litiumbatteri 18650</b>	Strömförsörjning
<b>Batterihållare för litiumbatteri 18650</b>	
<b>Grovekontakter 4-stift</b>	För modularisering av I <sup>2</sup> C-komponenter

## 2.3 Utveckling av stegräknaren

Två prototyper har byggts under projektets gång. Dessa beskrivs nedan.

### 2.3.1 Prototyp 1

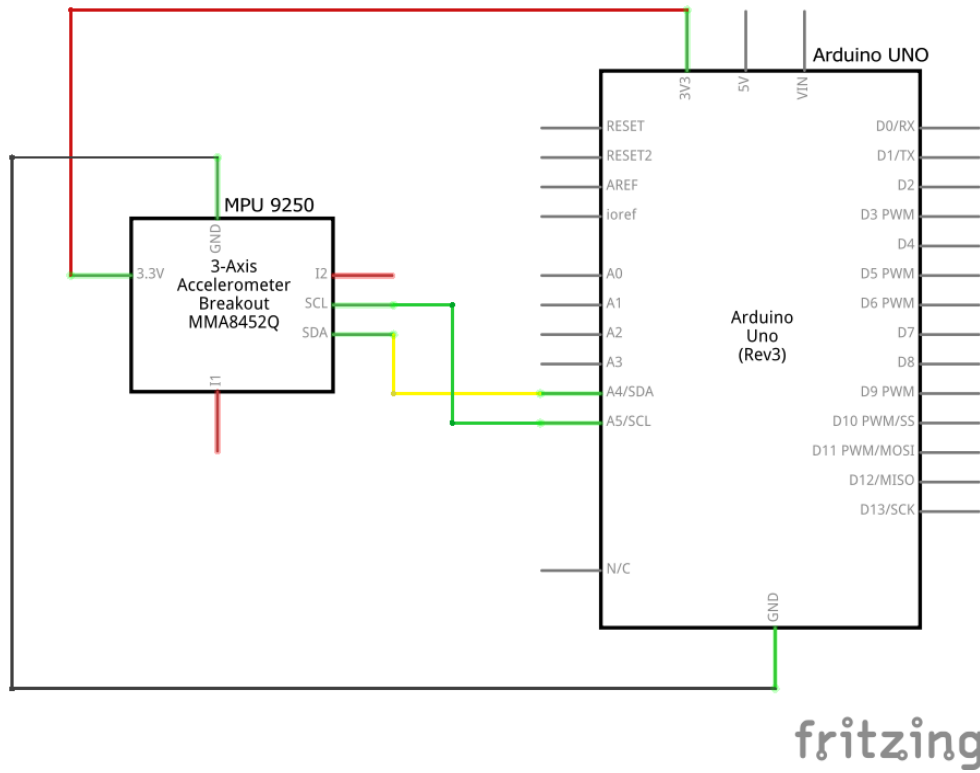
Den första prototypen byggdes med en Arduino UNO och en breadboard dit komponenter kopplades. Prototypen testades genom att stoppa ner den i en ficka och promenerades runt med. Antal steg skrevs ut till *Serial Monitor* som finns i Arduinos IDE. En bärbar dator användes för strömförsörjning via USB, och för att se datan från Serial Monitor.

Både en accelerometer (MPU-9250) och vibrationssensor testades. Accelerometern valdes att gå vidare med, då denna ansågs ge bäst och stabilast värden.

Utskrift av steg till serial monitor:



Kopplingschema för prototyp 1:

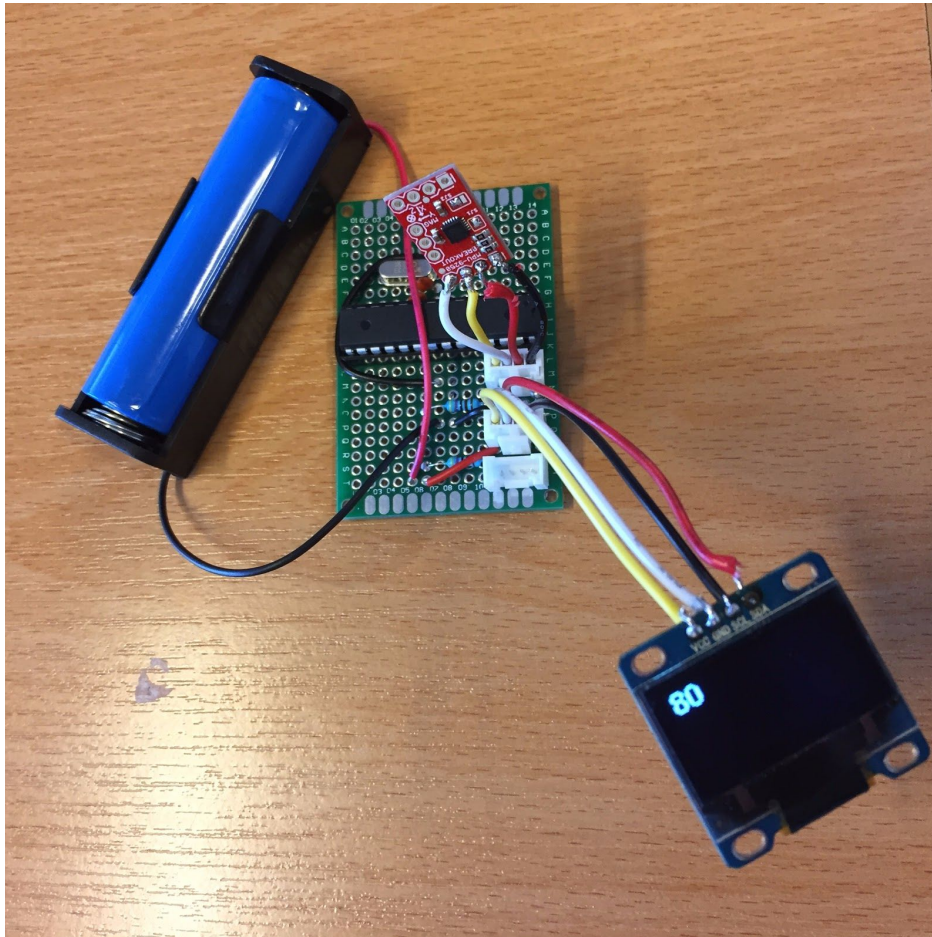




### 2.3.2 Prototyp 2

Till prototyp nummer 2 används en Atmega328p MCU (Micro controller unit) i stället för Arduino. Komponenter har löts samman på en platta och satts i ett hölje byggt av en optisk mus. Kontakter har använts för modularisering. Samtliga I2C-enheter går att byta ut eller placera om. Ett batteri används för strömförsörjning. Stegräknaren har minskat i storlek avsevärt sedan prototyp 1.

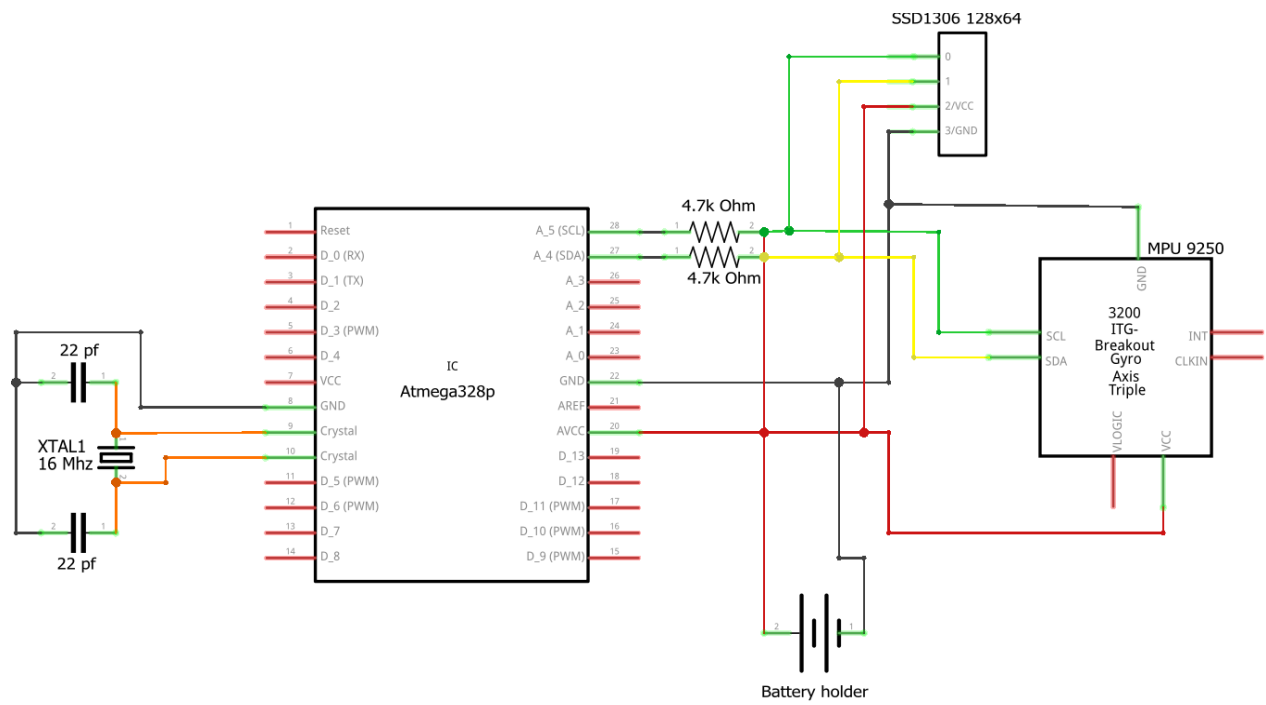
Fotografi på prototyp 2, utan hölje:



Fotografi på prototyp 2, med hölje:



## Kopplingschema för prototyp 2



fritzing

## 2.4 Stegräknarens funktion

Stegräknaren fungerar genom att hämta accelerationsdata från MPU-9250-enheten.

Värden för acceleration i riktningarna X, Y och Z adderas till ett enda värde. Detta sammansatta värde jämförs sedan mot ett värde som anses vara viloläge för accelerometern, dvs när den ligger helt stilla. Om skillnaden mellan dessa två värden överskrider ett visst satt tröskelvärde, så anses det att ett steg har tagits.

Vid varje taget steg uppdateras stegräknarens display med totalt antal tagna steg.

## 2.5 Kod

Koden för stegräknaren är skrivet i programspråket C. Redan färdiga kodbibliotek för displayen, accelerometern och I2C har använts för att underlätta programmeringsarbetet.

Utförligare kommentering av koden finns i källkodsfilen.

### 2.5.1 Använda icke-standardbibliotek

Namn	Författare	Beskrivning
<b>I2C master library using hardware TWI interface</b>	Peter Fleury	I2C-funktionalitet för AVR-enheter (Atmel MCUer)
<b>MPU6050 lib 0x02</b>	Davide Gironi	För accelerometer-enheten, via I2C.
<b>Universal 8bit Graphics Library</b>	olikraus@gmail.com	För display, via I2C.

### 2.5.2 Globala variabler

Följande globala variabler används i projektets kod.

Namn	Beskrivning
<b>uint16_t steps</b>	Håller antal steg
<b>double accIdle</b>	Värde som sätts vid programmets uppstart, håller det värde som accelerometern ger när den ligger helt stilla.
<b>double accCombined</b>	Sammansatt accelerationsvärde från X- Y- och Z-riktning.
<b>double accX, accY, accZ</b>	Värden för acceleration i X- Y- och Z-riktning.

### 2.5.3 Definitioner / macron

Följande macron används i projektets kod

Namn	Värde	Beskrivning
<b>X</b>	<b>0x3B</b>	Address för accelerometerdata i X-riktning i MPU
<b>Y</b>	<b>0x3D</b>	Address för accelerometerdata i Y-riktning i MPU
<b>Z</b>	<b>0x3F</b>	Address för accelerometerdata i Z-riktning i MPU
<b>STEP_ACC_TRIGGER</b>	<b>2.3</b>	Tröskelvärde för att bestämma om ett steg har tagits.

### 2.5.4 Funktioner

Beskrivning av egenskrivna funktioner i projektets kod

Namn	Beskrivning
<b>double getAcc(int addr);</b>	Hämtar accelerationsdata från MPU och ger värdet i retur, <b>angivet som g-kraft (SANT????)</b> . Som argument tar funktionen adressen till det register som ska läsas. Dessa adresser är definierade som macron <b>X</b> , <b>Y</b> och <b>Z</b> .
<b>void drawSteps(uint16_t steps);</b>	Skriver ut aktuellt antal steg till displayen.
<b>void drawString(char * string);</b>	Skriver ut en text till displayen. Texten ges som argument till funktionen, i form av en pekare till en char-sträng.
<b>void setAccIdle();</b>	Sätter ett värde till den globala variabeln <b>accIdle</b> . Funktionen körs endast en gång vid varje uppstart av programmet.
<b>double getAccXYZ(void);</b>	Använder sig av funktionen <b>getAcc</b> för att hämta accelerationsdata från <b>X</b> -, <b>Y</b> - och <b>Z</b> -riktningen. Det adderade värdet ges i retur.



## 3 Referenser

Länkar till författare av kodbibliotek

[MPU 6050 library - Davide Gironi](#)

[U8glib - Olikarus](#)

[I2C master library - Peter Fleury](#)

Länkar till använda datablad för komponenter

[MPU9250](#)

[Atmega328p](#)

[SSD1306 Oled Display 128x64](#)

## 4 Slutsats

Skrivs senare